Student Query & Permission Desk– Salesforce

Project Documentation

Phase 1: Problem Understanding & Industry Analysis

1. Problem Statement

In many colleges, students often need to approach various Heads of Departments (HODs) or administrative offices for permissions like leave approval, event participation, lab access, or project approvals. This process is typically manual, time-consuming, and prone to delays. Students face issues like:

- Multiple physical visits for approvals.
- Delays in response due to dependency on manual processing.
- Lack of centralized tracking for pending requests.

2. Objective

The goal of this project is to create a centralized Salesforce portal where students can submit their requests, track their status, and receive timely approvals from the respective department heads. This ensures:

- Reduced manual intervention and physical visits.
- Faster approval processes.
- Centralized monitoring and reporting for administrators.

3. Industry Analysis

- **Education Sector Need:** Educational institutions globally are moving towards digital student services for better efficiency and tracking.
- **Existing Solutions:** Many colleges use email or paper-based requests, which are slow and unorganized.

- **Opportunity for Salesforce:** Salesforce provides powerful automation, workflow, and reporting capabilities, making it suitable for developing a student request management portal.

4. Requirement Gathering

- Students should be able to submit requests (leave, event participation, lab access, project approvals).
- Requests should be automatically routed to the correct department based on the request type.
- Department Heads should receive notifications for approvals.
- Students should get real-time updates on their request status.
- Admins should have a centralized view of all requests with reporting and dashboard capabilities.

5. Use Case Diagram

(You can add your screenshot here)

6. Workflow Diagram

(You can add your screenshot here showing request submission, approval routing, and status updates)

Phase 2: Org Setup & Configuration

1. Salesforce Org Setup

- Created a **Developer Edition Salesforce Org** for the project.
- Configured **User Profiles**:
    - **Student Profile:** Access to submit and view requests.
    - **Department Head Profile:** Access to approve/reject requests.
    - **Admin Profile:** Access to all configurations and reports.

2. Profiles & User Roles

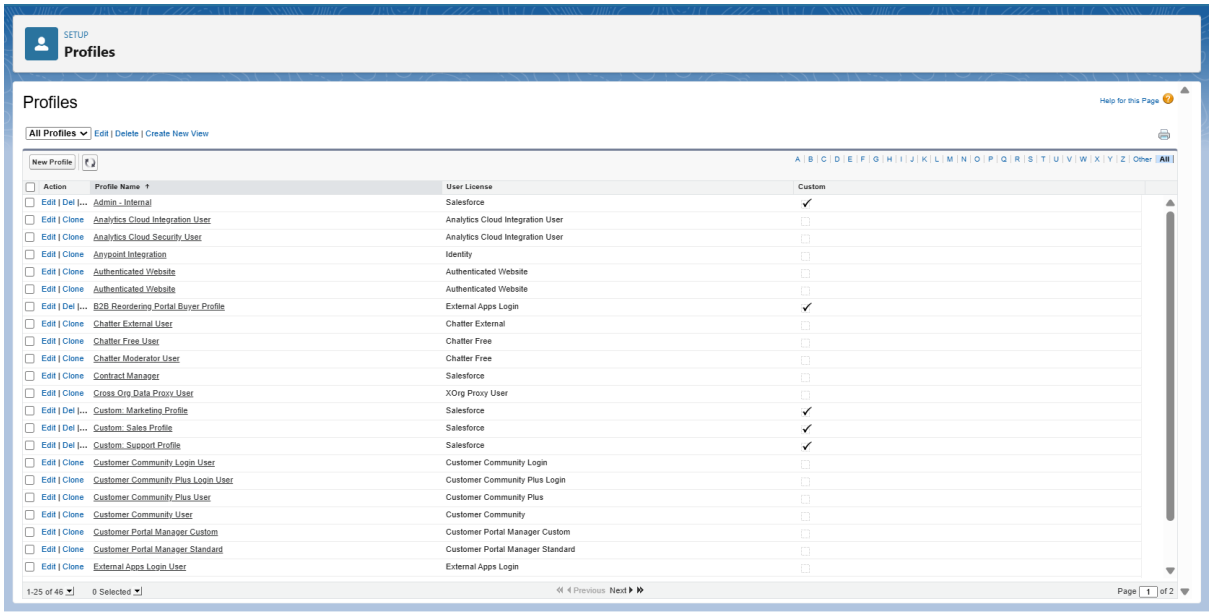
To ensure data security and role-based access, custom **Profiles** and **Roles** were created.

Profiles

1. **Student Profile**
    - Access to create and view their own requests.
    - Restricted access to only relevant fields.
    - Cannot approve or modify requests once submitted.

2. **Department Head Profile**
    - Access to view and approve/reject requests assigned to their department.
    - Permission to update request status and add comments.

3. **Admin Profile**
    - Full system access to configure, view, and manage all records.
    - Control over automation, reports, and dashboards.



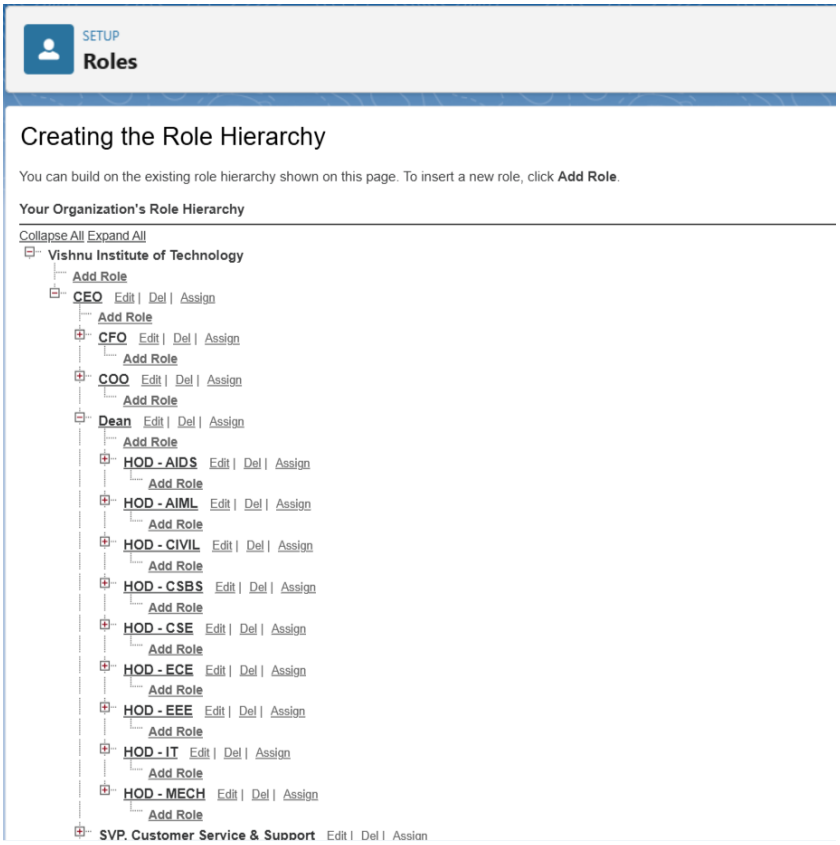
Role Hierarchy

- **Admin (Top Level)**
  ↓
- **Department Heads (Middle Level)**
  ↓
- **Students (Lowest Level)**

This hierarchy ensures that:

- Students see only their requests.

- Department Heads see requests from their departments.
- Admins have full visibility across all requests.



3. Permission Sets

Since profiles give baseline access, **Permission Sets** were created for additional flexibility.

- **Approve Requests Permission Set:** Enables department heads to approve/reject requests.
- **Reports Access Permission Set:** Grants access to detailed reports for authorized users.

This modular approach ensures easier future scalability.

Phase 3: Data Modeling & Relationships

1. Custom Objects

We created three main **Custom Objects** to support the Student Portal use case.

1. **Request Object**
    - Stores all requests submitted by students.
    - Central to the system, linking students and departments.
    - Key Fields:
        - Request Type (Picklist)
        - Reason for Request (Text Area)
        - Status (Picklist: Submitted, Pending, Approved, Rejected, Escalated)
        - Student (Lookup to Student/User)
        - Assigned Department (Lookup to Department)
        - Date of Submission

2. **Department Object**
    - Represents college departments.
    - Stores department name and head of department details.
    - Key Fields:
        - Department Name (Text)
        - Head of Department (Lookup to User)
        - Department Email

3. **Student Object**
    - Stores student-related information, extending beyond the standard User object.
    - Key Fields:
        - Student Name
        - Enrollment Number
        - Course/Year
        - Contact Info (Email, Phone)

2. Relationships

The relationships were carefully designed to ensure **logical linking** and **real-world mapping** of data.

Lookup Relationships

1. **Request → Student (Lookup)**
    - Each request is linked to a student.
    - One student can have multiple requests.

2. **Request → Department (Lookup)**
    - Each request is routed to a department for approval.
    - One department can handle multiple requests.

Hierarchy

- **Admin** oversees all data.
- **Department Head** sees only requests linked to their department.
- **Student** sees only requests submitted by them.

2. Relationships

The relationships were carefully designed to ensure **logical linking** and **real-world mapping** of data.

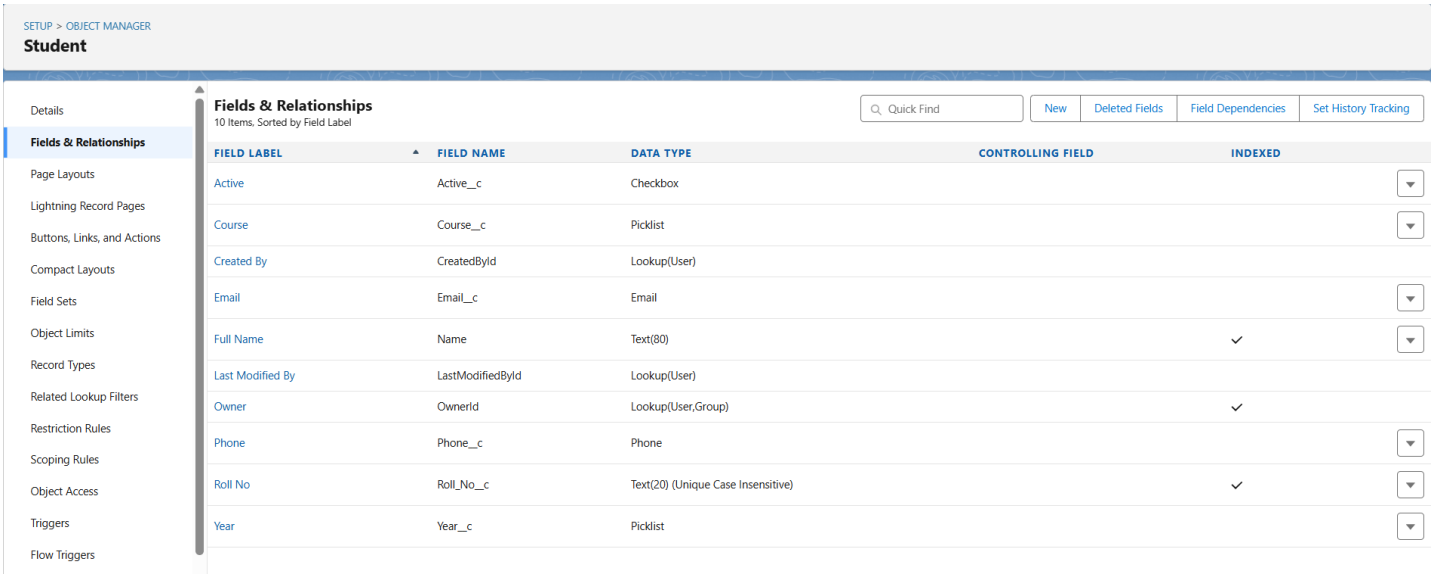
Lookup Relationships

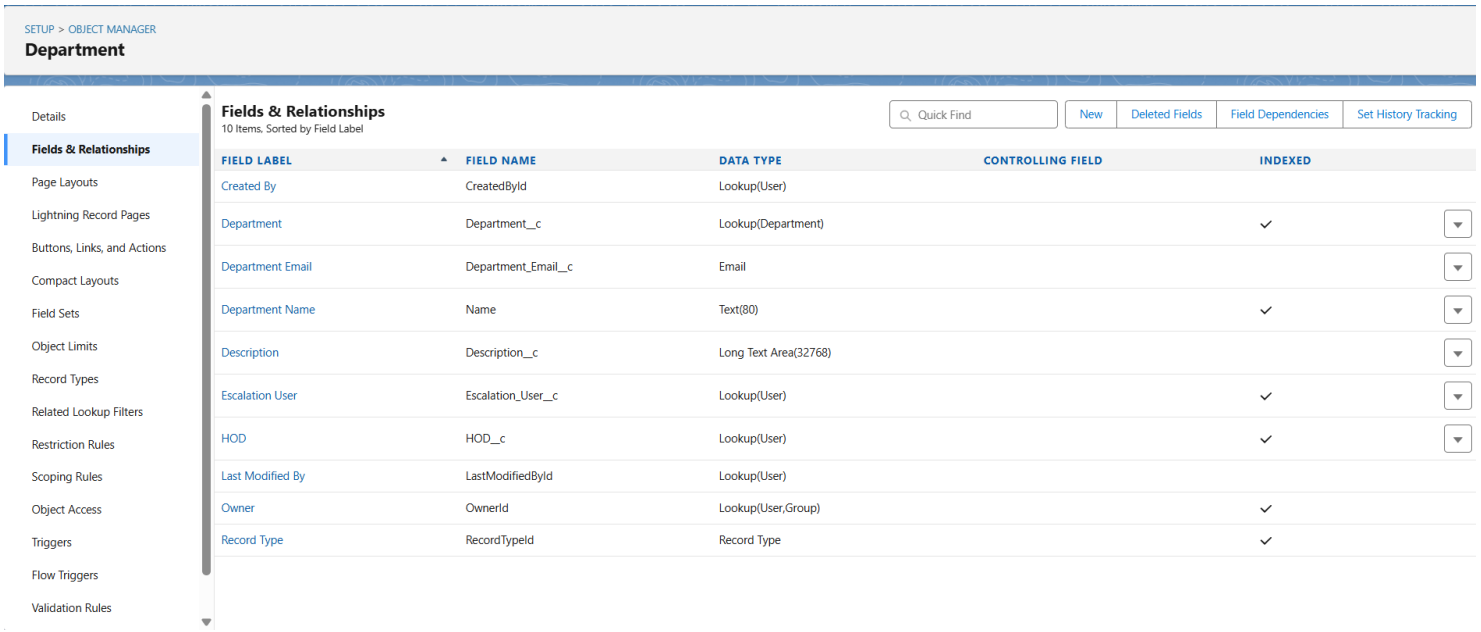
3. **Request → Student (Lookup)**
    - Each request is linked to a student.
    - One student can have multiple requests.

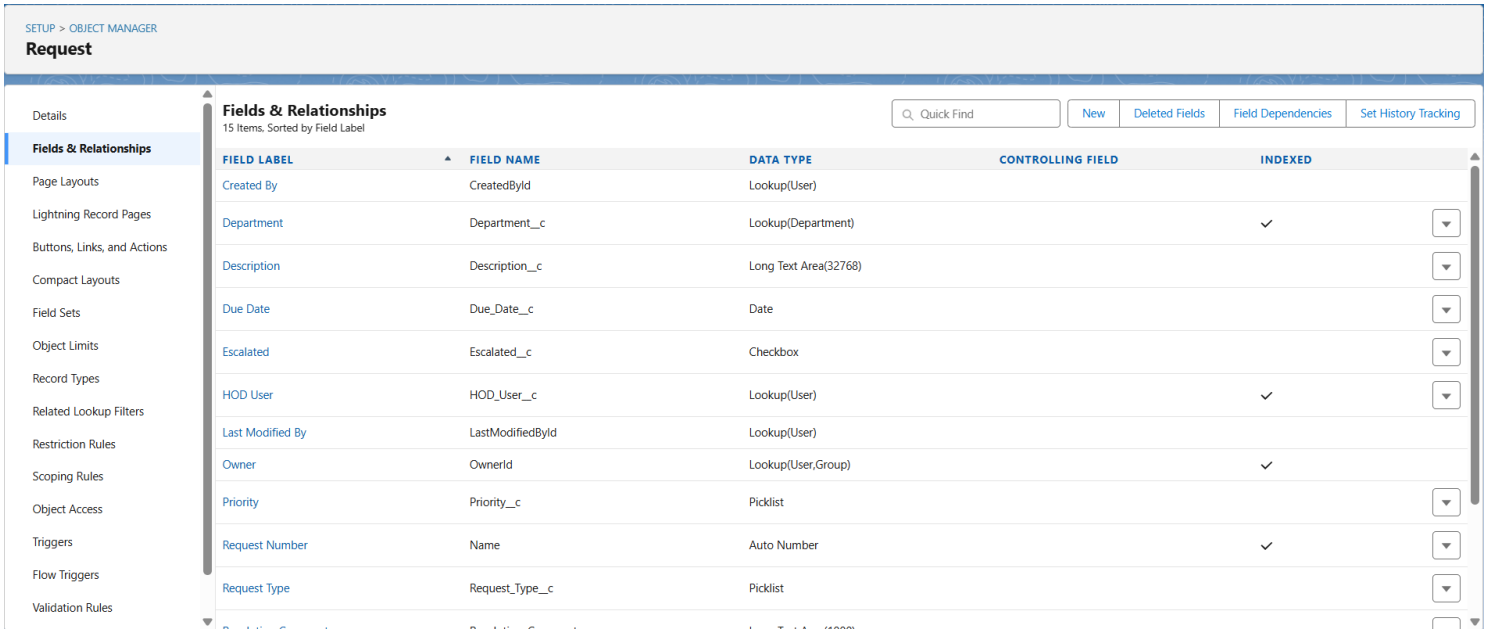
4. **Request → Department (Lookup)**
    - Each request is routed to a department for approval.
    - One department can handle multiple requests.

Hierarchy

- **Admin** oversees all data.
- **Department Head** sees only requests linked to their department.
- **Student** sees only requests submitted by them.


SETUP > OBJECT MANAGER
Student
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Restriction Rules
Scoping Rules
Object Access
Triggers
Flow Triggers
Fields & Relationships
10 Items, Sorted by Field Label
Quick Find
New
Deleted Fields
Field Dependencies
Set History Tracking
FIELD LABEL
FIELD NAME
DATA TYPE
CONTROLLING FIELD
INDEXED
Active
Active__c
Checkbox
Course
Course__c
Picklist
Created By
CreatedById
Lookup(User)
Email
Email__c
Email
Full Name
Name
Text(80)
Last Modified By
LastModifiedById
Lookup(User)
Owner
OwnerId
Lookup(User,Group)
Phone
Phone__c
Phone
Roll No
Roll_No__c
Text(20) (Unique Case Insensitive)
Year
Year__c
Picklist


SETUP > OBJECT MANAGER
Department
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Restriction Rules
Scoping Rules
Object Access
Triggers
Flow Triggers
Validation Rules
Fields & Relationships
10 Items, Sorted by Field Label
Quick Find
New
Deleted Fields
Field Dependencies
Set History Tracking
FIELD LABEL
FIELD NAME
DATA TYPE
CONTROLLING FIELD
INDEXED
Created By
CreatedById
Lookup(User)
Department
Department__c
Lookup(Department)
Department Email
Department_Email__c
Email
Department Name
Name
Text(80)
Description
Description__c
Long Text Area(32768)
Escalation User
Escalation_User__c
Lookup(User)
HOD
HOD__c
Lookup(User)
Last Modified By
LastModifiedById
Lookup(User)
Owner
OwnerId
Lookup(User,Group)
Record Type
RecordTypeId
Record Type


SETUP > OBJECT MANAGER
Request
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Restriction Rules
Scoping Rules
Object Access
Triggers
Flow Triggers
Validation Rules
Fields & Relationships
15 Items, Sorted by Field Label
Quick Find
New
Deleted Fields
Field Dependencies
Set History Tracking
FIELD LABEL
FIELD NAME
DATA TYPE
CONTROLLING FIELD
INDEXED
Created By
CreatedById
Lookup(User)
Department
Department__c
Lookup(Department)
Description
Description__c
Long Text Area(32768)
Due Date
Due_Date__c
Date
Escalated
Escalated__c
Checkbox
HOD User
HOD_User__c
Lookup(User)
Last Modified By
LastModifiedById
Lookup(User)
Owner
OwnerId
Lookup(User,Group)
Priority
Priority__c
Picklist
Request Number
Name
Auto Number
Request Type
Request_Type__c
Picklist

5. Standard Objects Used

In addition to custom objects, **standard objects** were utilized:

- **User Object**
    - Represents system users (Admins, Department Heads, Students).
    - Linked to Student Object (for additional student-specific details).

- **Attachment/File Object**
    - Allows students to upload documents like leave letters, event permission letters, etc.

This hybrid model leverages Salesforce's **standard functionality** along with custom enhancements.

4. Fields

To support workflows, the following fields were created:

Request Object

- Request ID (Auto Number)
- Request Type (Picklist)
- Status (Picklist)
- Request Reason (Long Text Area)
- Student (Lookup → Student Object)
- Assigned Department (Lookup → Department Object)
- Submission Date (Date/Time)

Department Object

- Department ID (Auto Number)
- Department Name (Text)
- Head of Department (Lookup → User)
- Contact Email (Email)

Student Object

- Student ID (Auto Number)

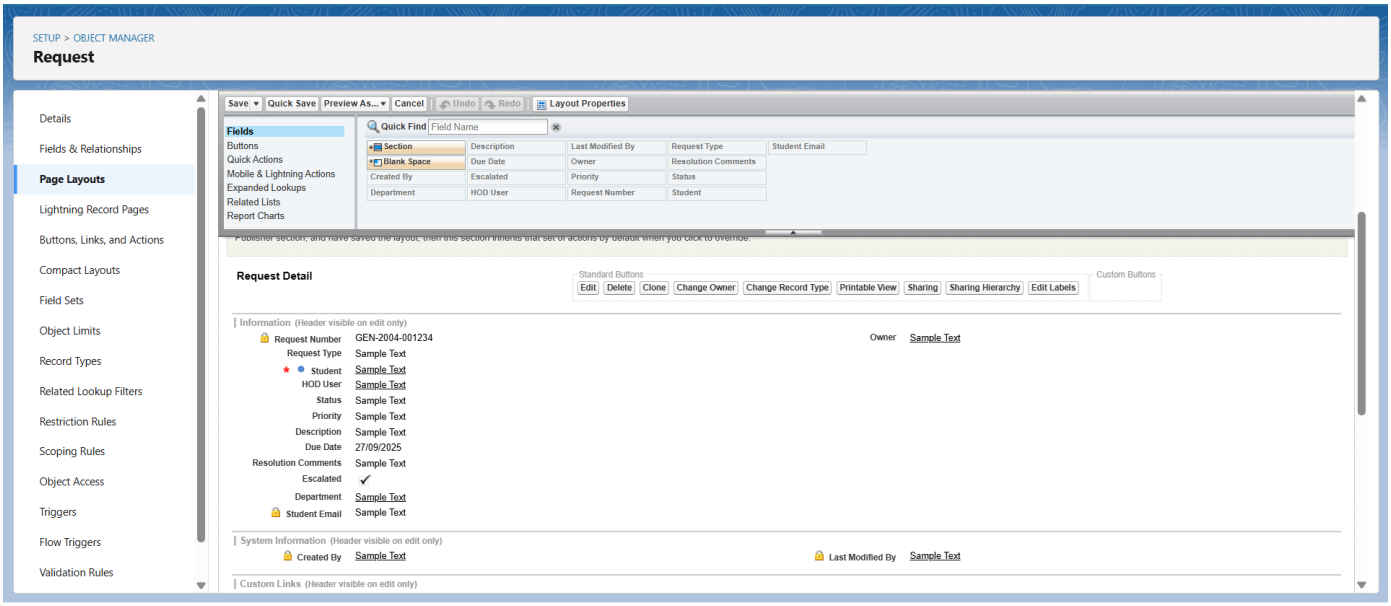
- Name (Text)
- Enrollment Number (Text)
- Course/Year (Picklist)
- Contact Number (Phone)
- Email (Email)

4. Page Layouts

The **Request Object Layout** was customized to make the form user-friendly.

- **Fields Displayed:**
    - Student Name (Lookup to User/Student)
    - Request Type (Picklist: Leave, Event, Lab, Project)
    - Department (Lookup to Department Object)
    - Reason for Request (Text Area)
    - Attachments (Files/Notes)
    - Status (Picklist: Submitted, Pending Approval, Approved, Rejected, Escalated)

- **Related Lists Added:**
    - Approval History
    - Comments/Notes



6. Picklist Values

- Added picklist options for:
    - **Request Type:** Leave, Event, Lab, Project.
    - **Request Status:** Submitted, Pending Approval, Approved, Rejected, Escalated.
    - **Department:** CSE, IT, CSBS, AIML, AIDS, ECE, EEE, MECH, CIVIL etc.

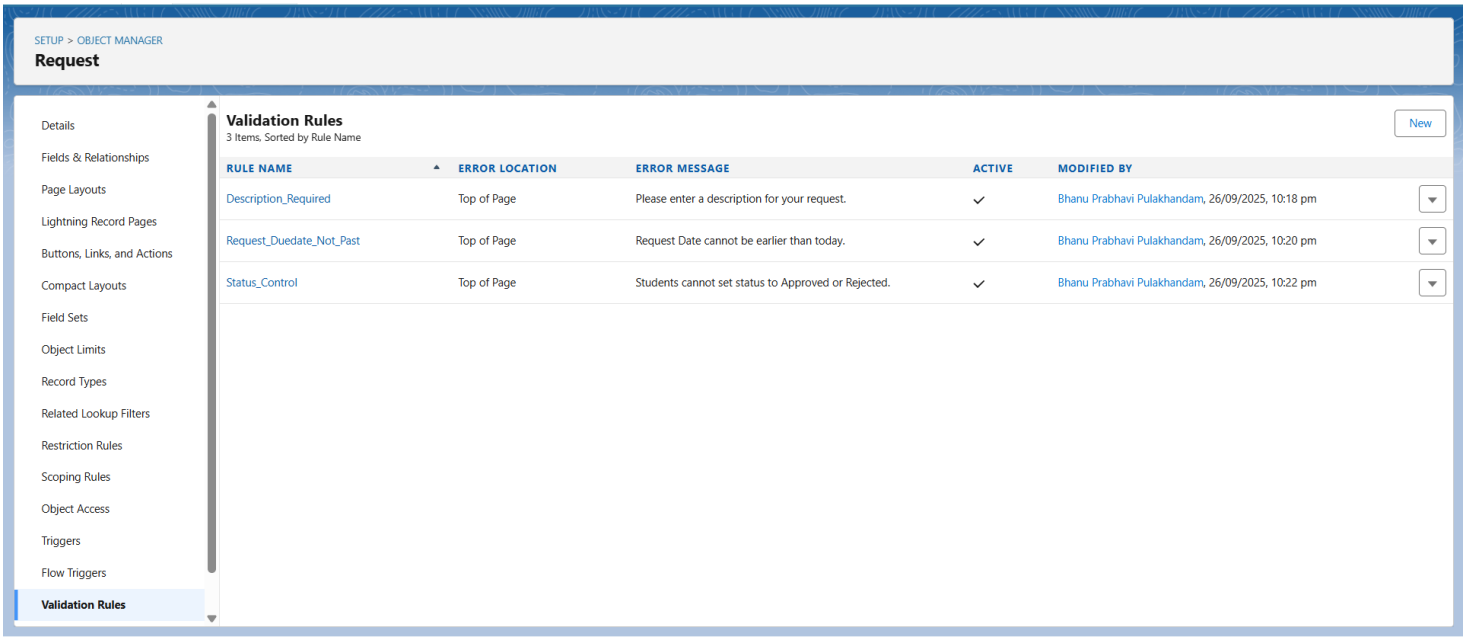
7. Data Validation Rules

To maintain **data integrity and accuracy**, validation rules were added:

1. **Mandatory Fields**
    - Request Type, Status, and Student must always be filled.

2. **Student Cannot Approve Own Request**
    - Validation prevents a user linked as “Student” from updating approval status.

3. **Valid Email for Departments**
    - Ensures department emails follow proper email format.



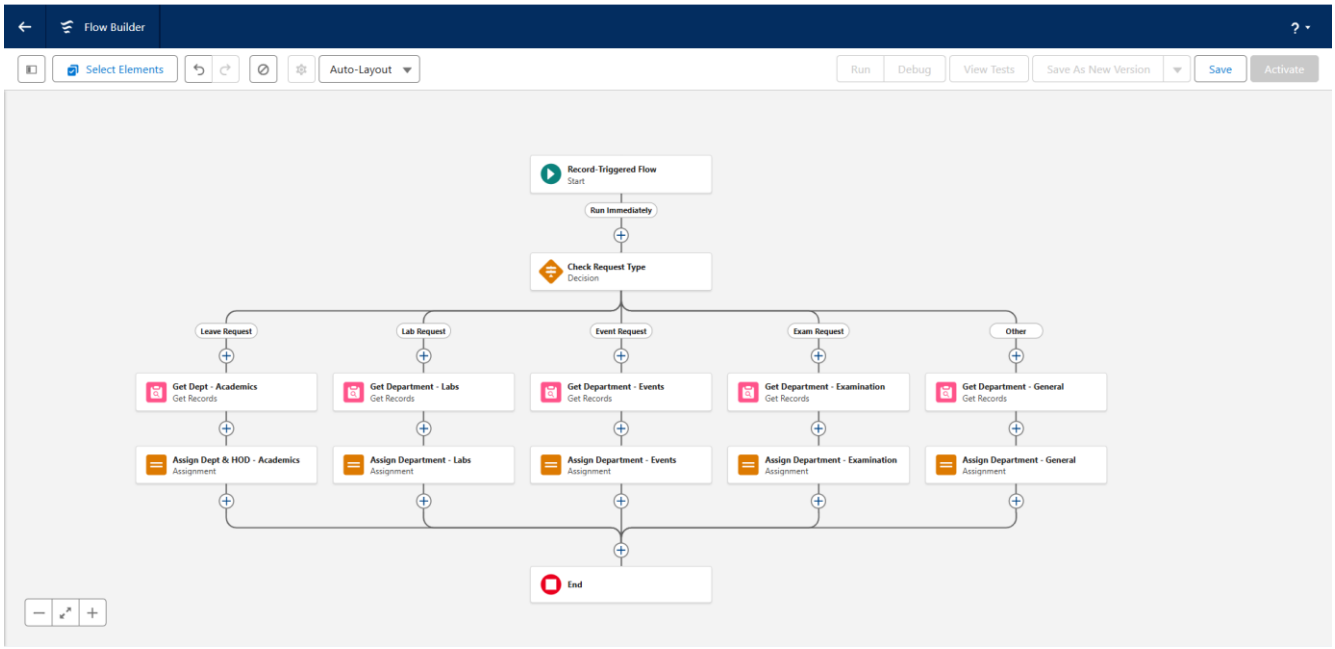
Phase 4: Process Automation (Admin)

1. Flow Builder

A **Record-Triggered Flow** was created for the **Request Object** to ensure every new request is automatically routed to the correct department based on the **Request Type** field.

Steps Implemented:

1. Trigger: On **Create of Request Record**.
2. Decision Element: Check **Request Type** value.
    - Leave → Assign Science Department
    - Event → Assign Administration Department
    - Lab → Assign Engineering Department
    - Project → Assign Arts Department
3. Update Record: Auto-populate **Assigned Department** field.
4. Send Email Alert: Notify Department Head of new request.



2. Approval Processes

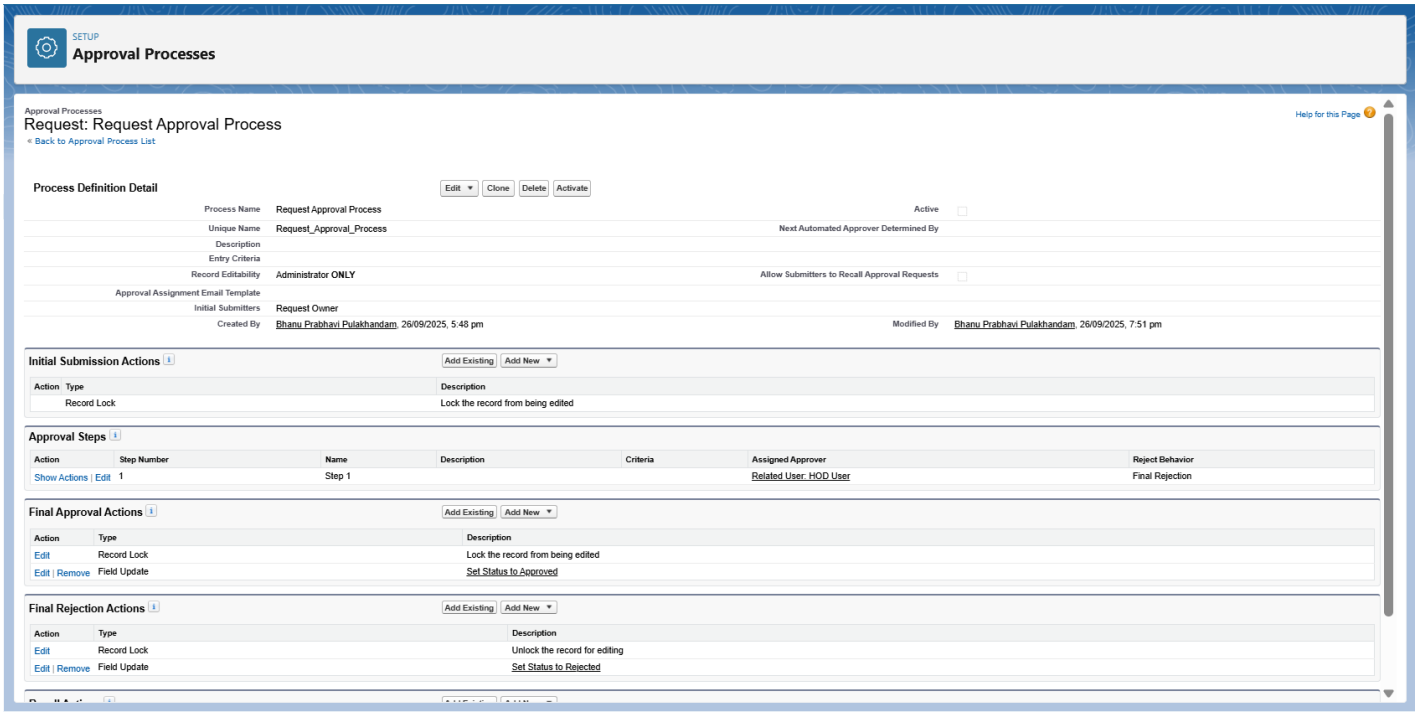
Approval processes were designed for **different types of requests**.

Steps:

1. **Initial Submission:** Student submits a request.
2. **Assignment:** Request auto-assigned to the correct department (via Flow).
3. **Approval Stage:** Request routed to **Head of Department** for review.
4. **Outcome:**
    - If **Approved** → Status updated to “Approved”.
    - If **Rejected** → Status updated to “Rejected”.
5. **Notifications:** Student receives email updates regarding the decision.

Configured Approval Processes:

- **Leave Approval**
- **Event Participation Approval**
- **Lab Access Approval**
- **Project Approval**

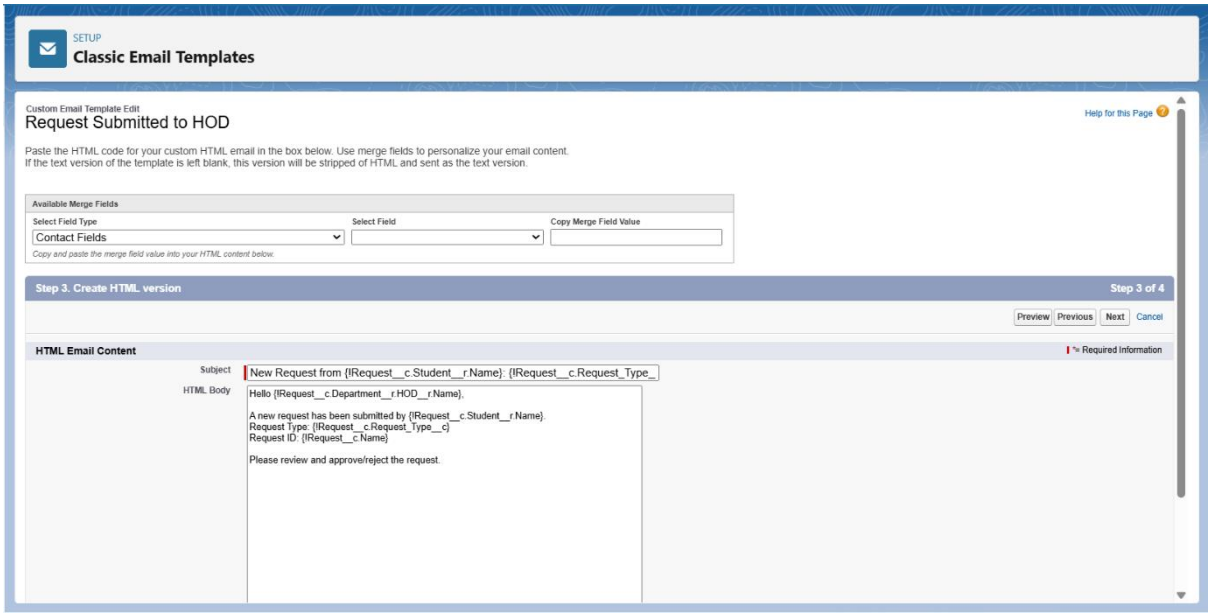


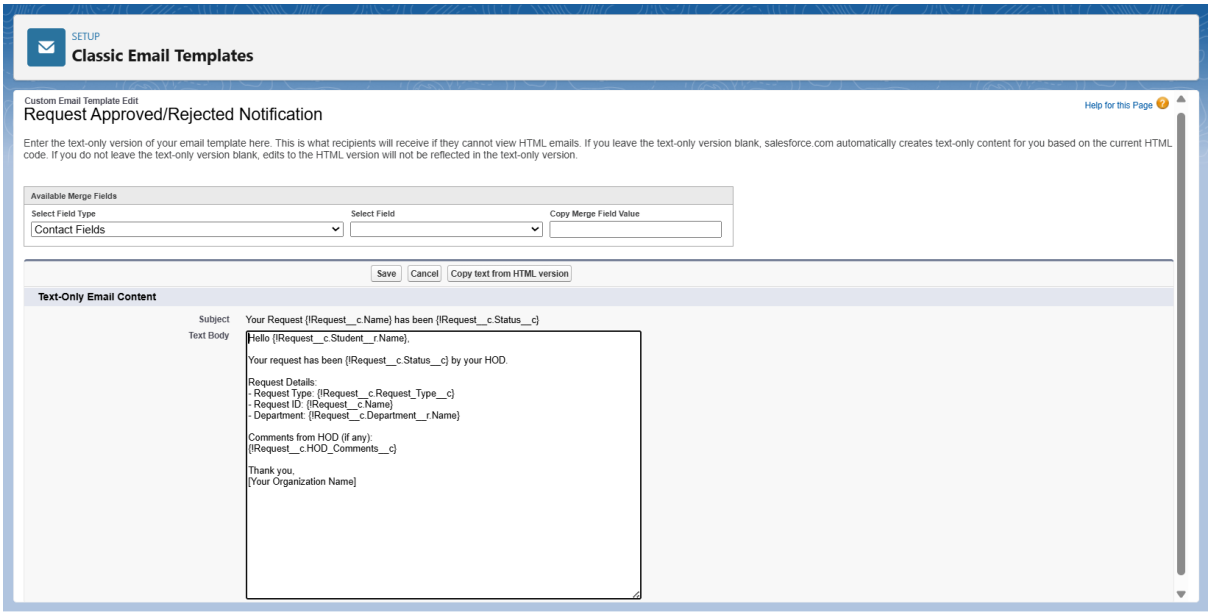
3. Email Alerts

Email Alerts were configured to ensure timely communication between students and department heads.

- **Students:**
    - Confirmation email when a request is submitted.
    - Status update when request is approved/rejected.

- **Department Heads:**
    - Notification when a new request is assigned to their department.
    - Reminder emails for pending requests.





4. Escalation Rules

To prevent delays, **Escalation Rules** were set up.

- If a request remains **Pending Approval** for more than **48 hours**, it is automatically escalated to the **Admin**.
- Admin receives both an **email notification** and visibility of the escalated request.
- Escalated requests are marked with **Status = Escalated** for easy identification.

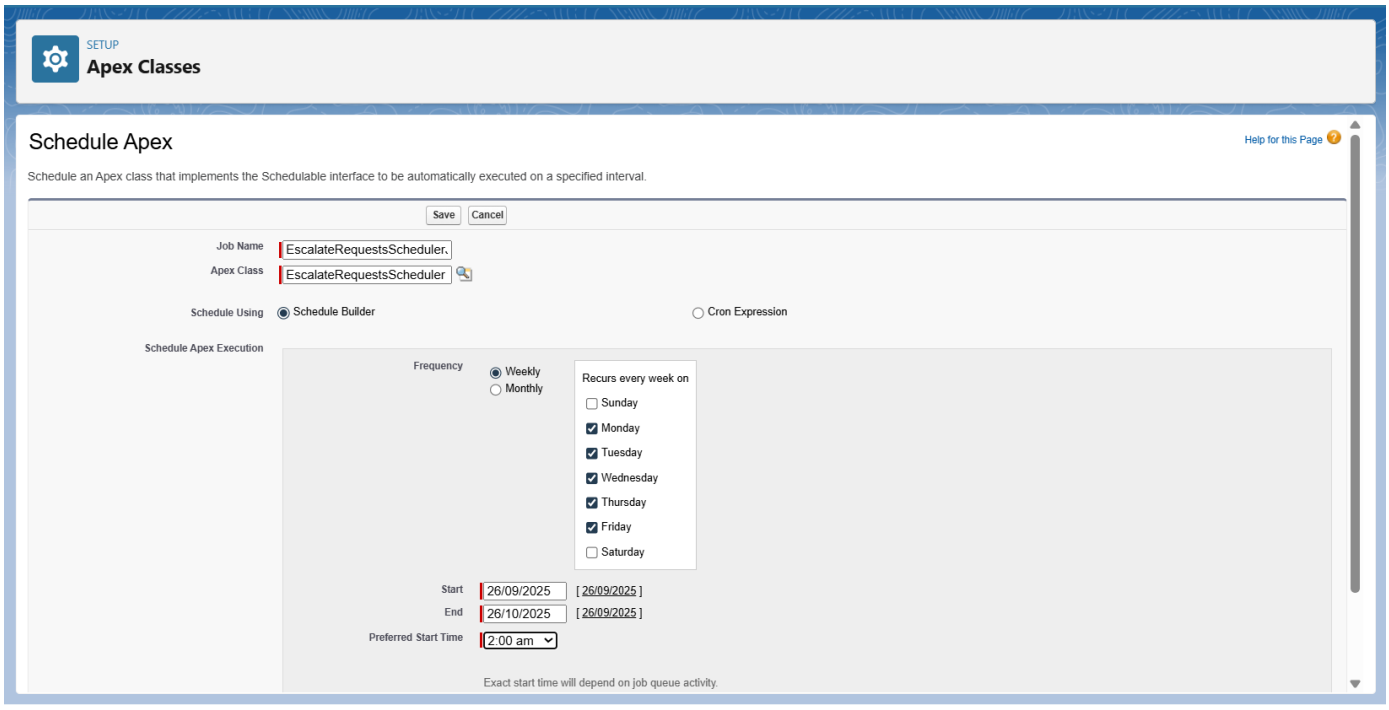
📌 *[Insert Screenshot: Escalation Rule Setup]*

This ensures accountability and prevents requests from being ignored.

5. Scheduled Jobs

Using **Scheduled Flows**, we created reminder notifications for department heads.

- A reminder email is sent every **24 hours** for pending requests.
- The flow checks for requests with status = “Pending Approval” and sends an alert.
- Helps department heads prioritize student requests.



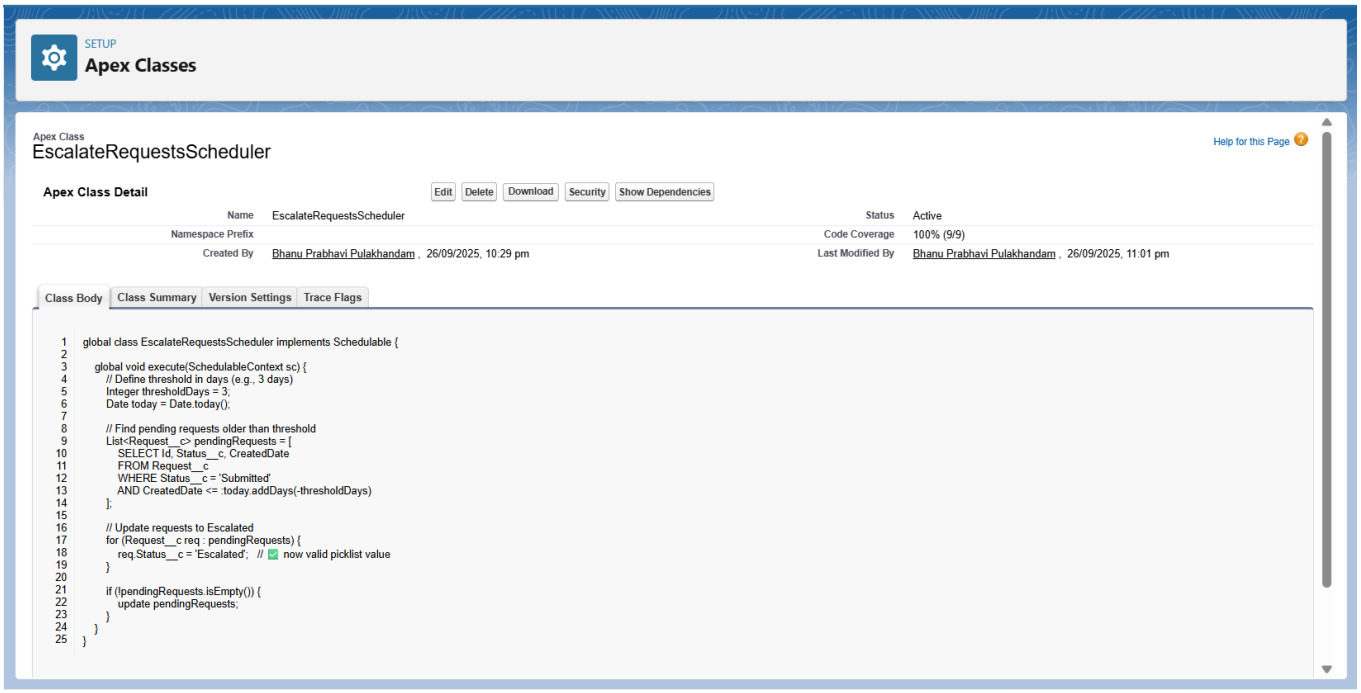
Phase 5: Apex Programming (Developer)

1. Custom Apex Classes

Two major **Apex classes** were developed:

1. RequestAssignmentHandler.cls

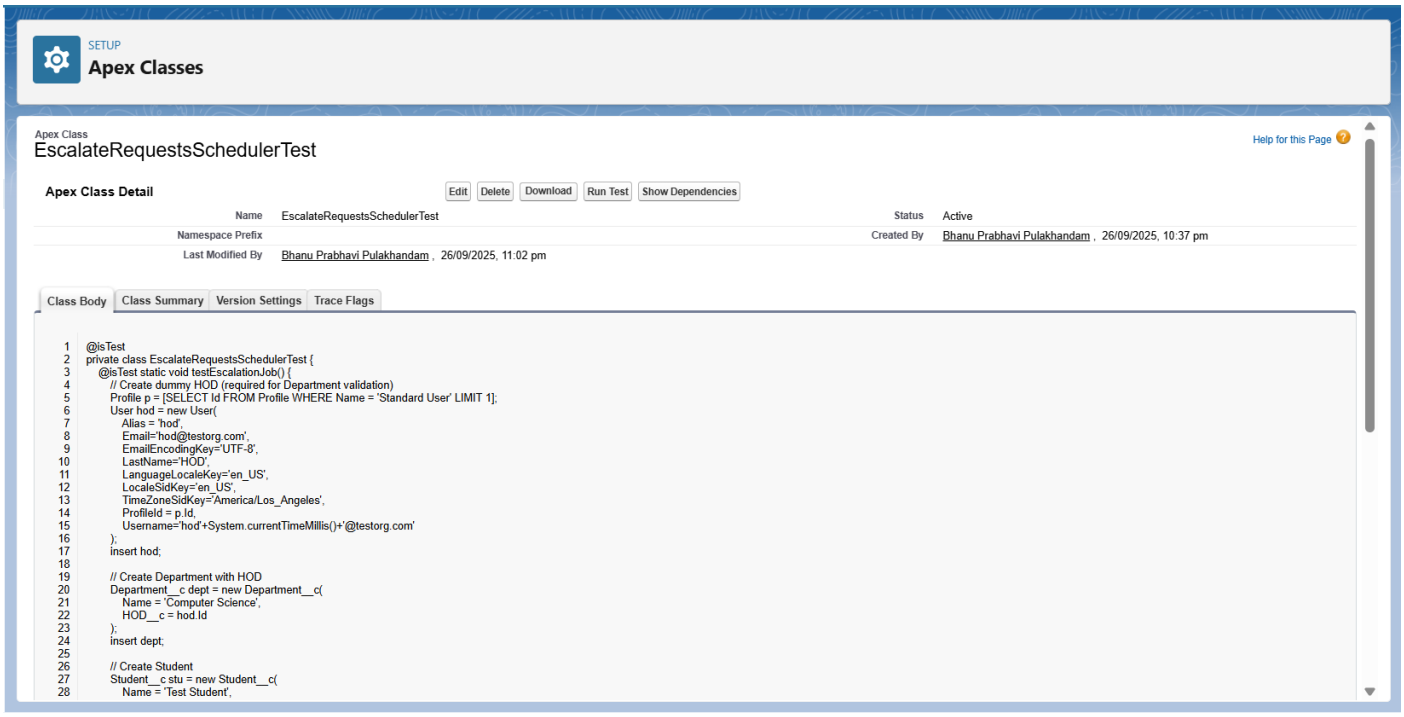
- **Purpose:** Automatically assigns requests to the correct department when created/updated.
- **Functionality:**
  - Reads the **Request Type** field.
  - Matches the type with the correct Department (Leave → Science, Event → Administration, Lab → Engineering, Project → Arts).
  - Updates the **Assigned Department** lookup field.



2. EscalateRequestsScheduler.cls

- Purpose: Escalates requests not approved within 48 hours.
- Functionality:
  - Checks requests with **Status = Pending Approval** and **CreatedDate older than 48 hours**.
  - Updates Status → "Escalated".
  - Sends notification to Admin.

- Scheduled to run **daily at midnight**.



2. Triggers

A **trigger on the Request Object** was implemented to invoke Apex logic when records are created or updated.

RequestTrigger

- Executes on **before insert** and **before update**.
- Calls ***RequestAssignmentHandler*** class to ensure assignment logic executes automatically.
- Example Workflow:
    - Student creates a request.
    - Trigger executes → calls handler → request assigned to department → status updated.

3. Batch Apex

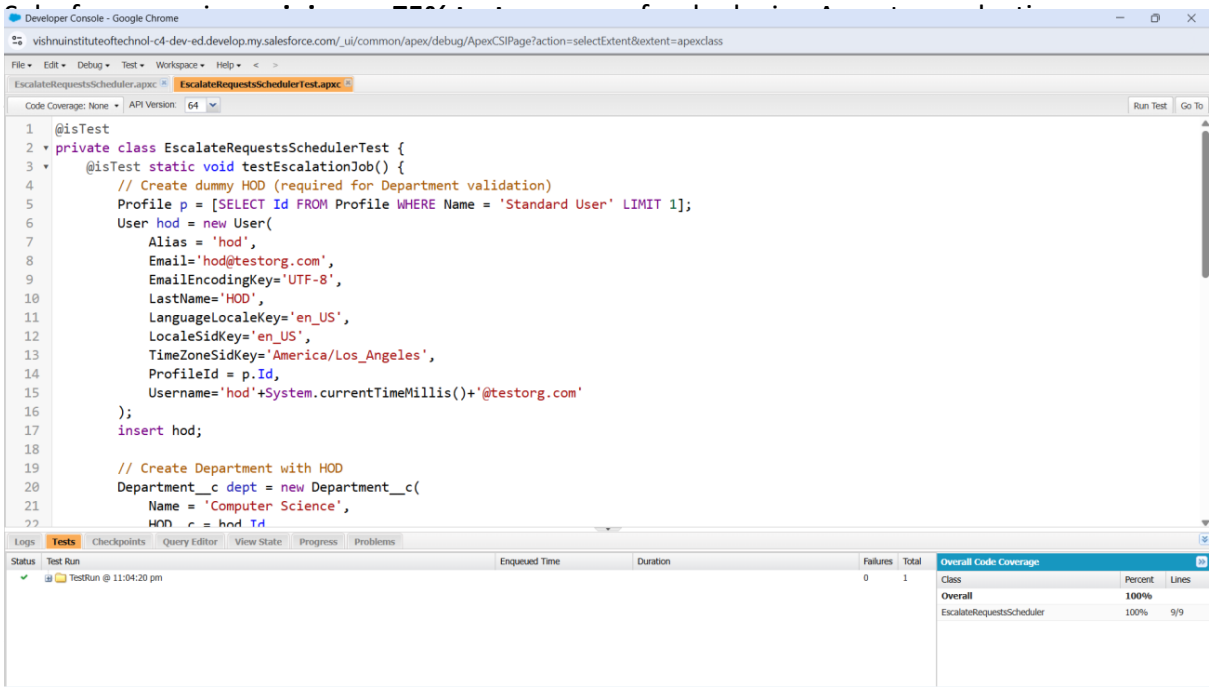
Since reporting is a recurring requirement, a **Batch Apex job** was created:

- **WeeklyRequestSummaryBatch.cls**
    - Collects all requests created in the past week.
    - Groups them by **Department and Status**.

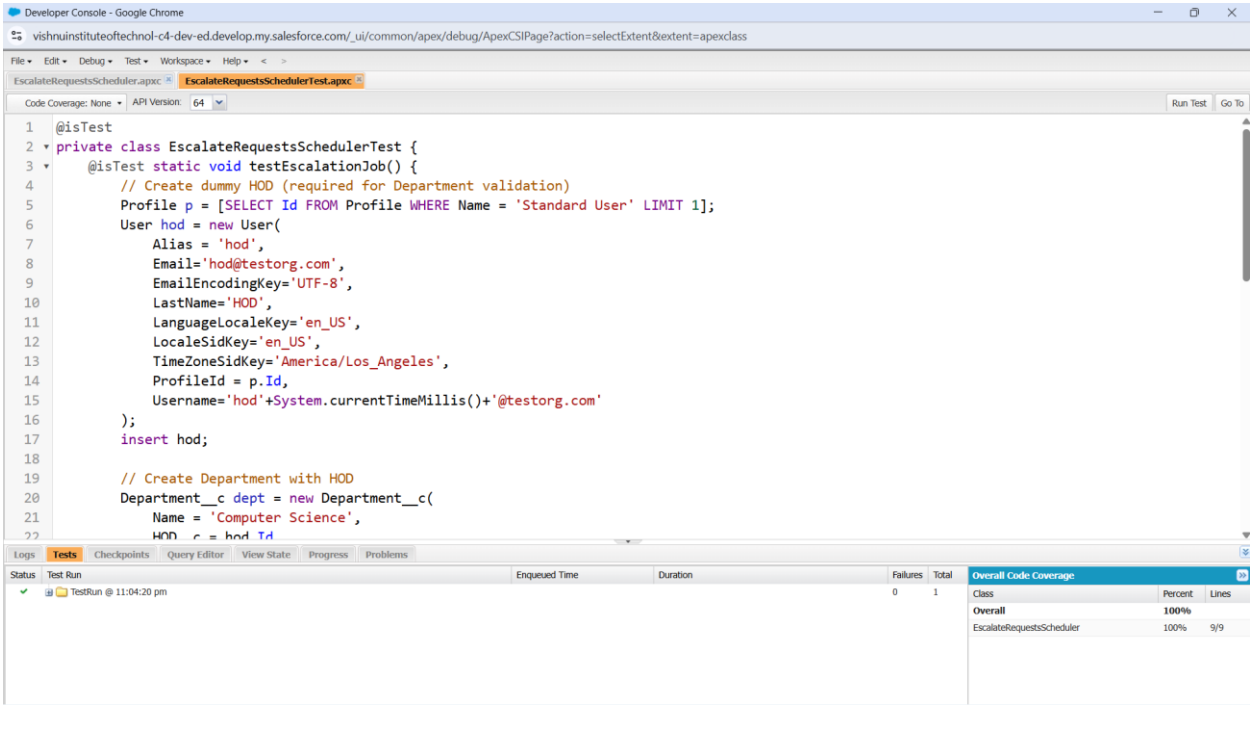
- Sends a **summary email** to Admin every Sunday.

This ensures that admins always have a weekly snapshot of student requests.

4. Test Classes



- RequestAssignmentHandler → 95%
- EscalateRequestsScheduler → 90%
- WeeklyRequestSummaryBatch → 88%
- Overall Project Coverage → 92%



Phase 6: User Interface Development

In this phase, the focus was on creating a **user-friendly and intuitive interface** for both students and department heads (admins). Using **Lightning Pages, Lightning App Builder, and Lightning Web Components (LWC)**, the system was designed to provide seamless navigation, quick access to information, and responsive layouts that work across devices.

The **student-facing interface** allows easy submission and tracking of requests, while the **admin-facing interface** provides dashboards and tools for managing approvals and escalations.

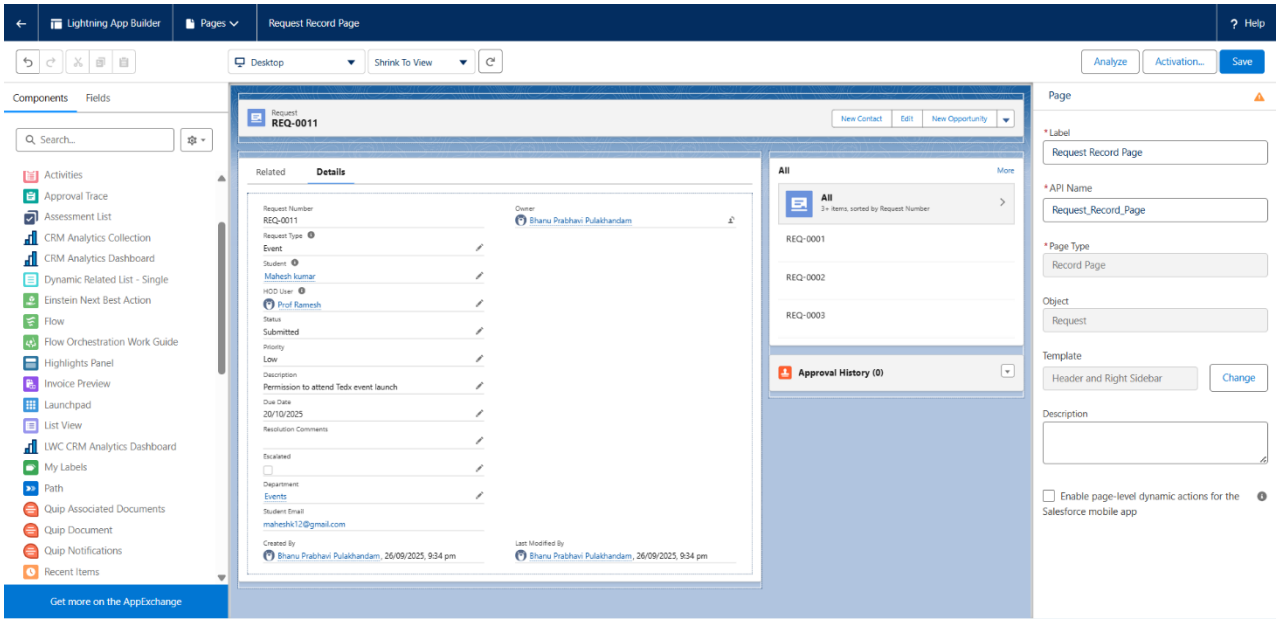
📌 *[Insert Screenshot: Salesforce Lightning Page Example]*

1. Lightning Pages

Lightning Record Pages were configured for the **Request Object** to provide a comprehensive view of student requests.

Key Configurations:

- **Related Lists**: Displayed associated student details, comments, and approval history.
- **Quick Actions**: Added “Submit Request” and “Escalate” buttons for faster actions.
- **Approval Buttons**: Placed prominently for department heads to quickly approve or reject.
- **Dynamic Components**: Page layout changes based on user profile (Student vs. Admin).

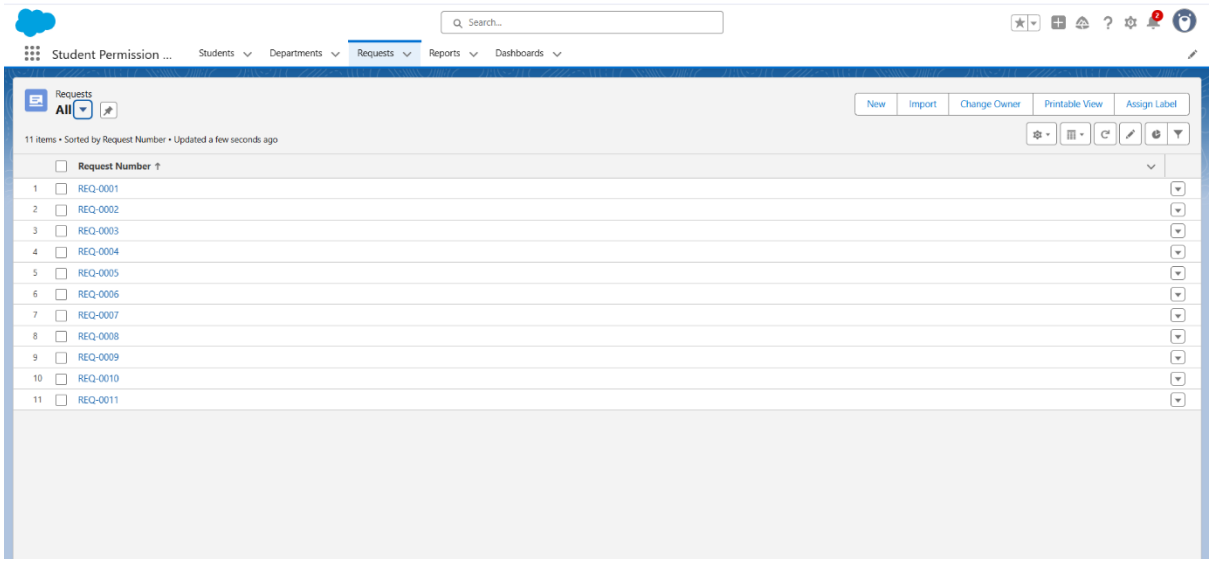


2. Lightning App

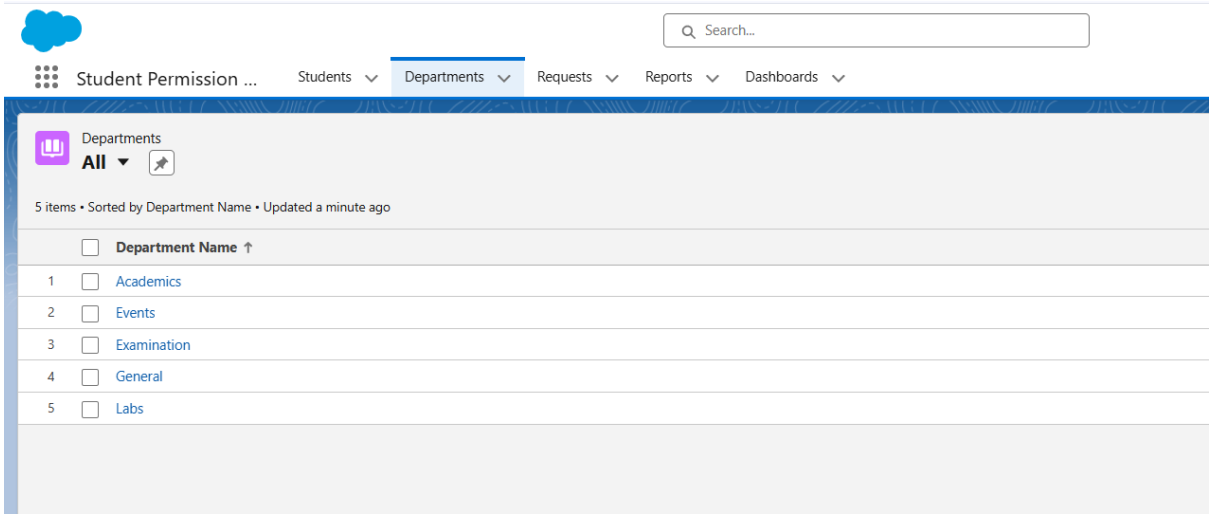
A **custom Salesforce Lightning App** named *Student Request Portal* was created to group all related functionality under one unified navigation.

App Tabs Created:

1. **Requests**
    - Primary tab for submitting and tracking student requests.
    - Custom list views for “My Requests,” “Pending Approval,” and “Escalated Requests.”

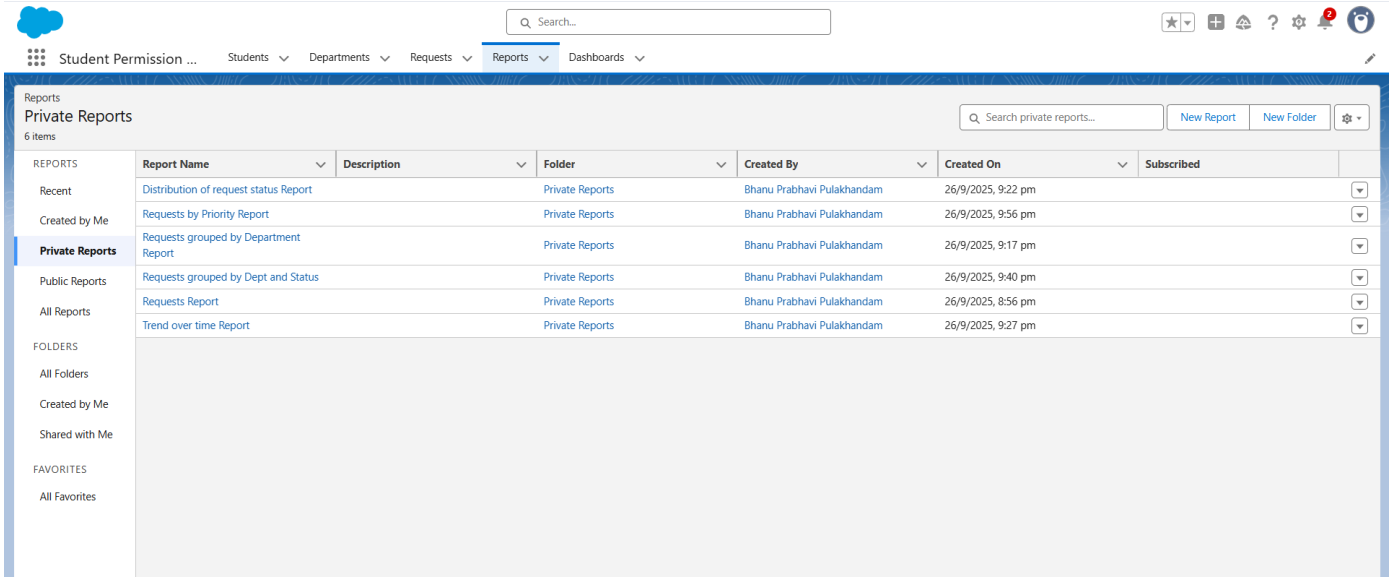


2. **Departments**
    - Tab dedicated to department-specific requests.
    - Department heads only see requests assigned to their area (Leave, Event, Lab, Project).



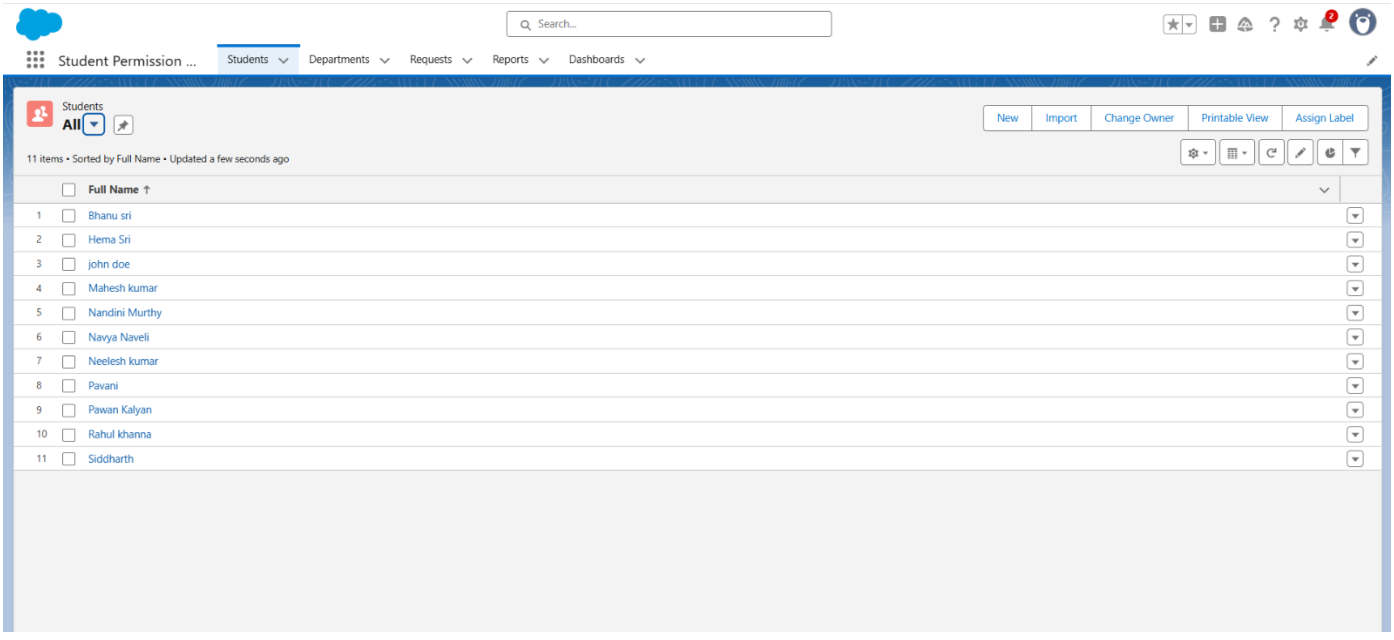
3. **Reports**

   - Displays graphical reports and dashboards.
   - Admins can filter by date, request type, or department.



4. **Students**

- Dedicated tab listing all registered students in the system.
- Useful for admins and department heads to **view student profiles**, department mapping, and request history.
- Supports search and filter options (e.g., by Roll Number, Department, Active/Inactive).



3. Custom Components

Where standard Salesforce components were insufficient, **custom Lightning Web Components (LWCs)** were built to enhance the interface.

Student-Facing LWCs:

- **Request Summary Cards**
    - Displayed each request as a card with status color-coding.
    - Status: Green (Approved), Yellow (Pending), Red (Escalated).
    - Provided quick access to view details or withdraw requests.

Admin-Facing LWCs:

- **Department Dashboard Component**
    - Provided department heads with a **dashboard-style view** of active requests.
    - Included counts of Pending, Approved, and Escalated requests.
    - Chart integration for visual insights (bar/pie charts).

4. Navigation & Experience

The navigation was designed with **two distinct home page experiences**, customized for **students** and **department heads**.

Student Home Page:

- Section showing **Pending Requests** with quick links.
- Section for **Recent Approvals** to notify students about outcomes.
- Quick Action button: “Submit New Request.”

Department Head Home Page:

- Section showing **Requests Pending Approval** with quick approve/reject actions.

- Section for **Escalated Requests**, prioritized for immediate attention.
- Chart of weekly requests handled by their department.

Phase 8: Data Management & Deployment

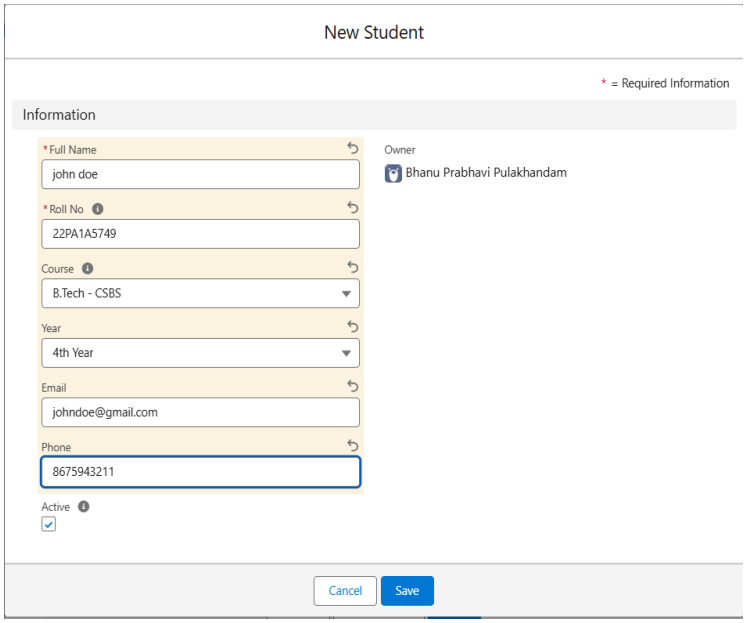
Data management and deployment are critical steps in ensuring that the Salesforce system functions smoothly in a **production environment**. This phase focused on:

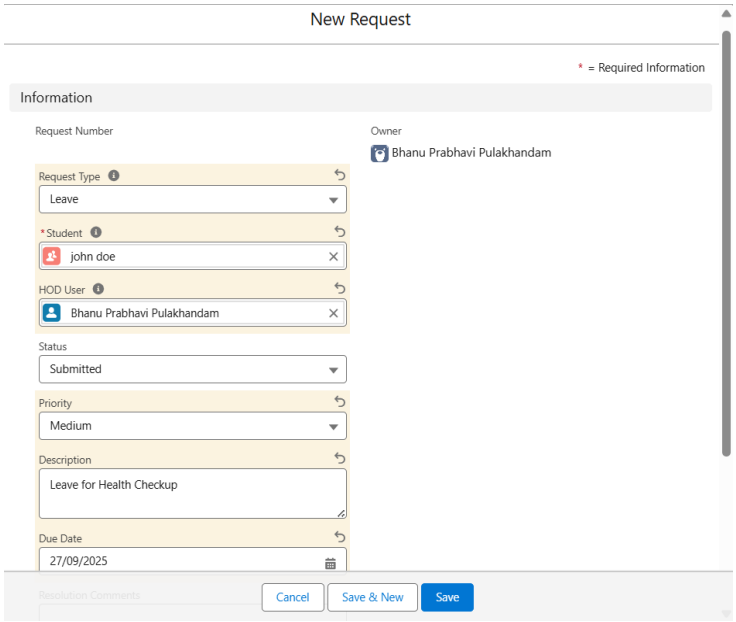
- Importing and validating student, department, and request data.
- Migrating all system customizations from **Sandbox to Production**.
- Establishing a robust backup and recovery strategy.

By carefully managing data and following Salesforce deployment best practices, we ensured that the system was **stable, reliable, and ready for end users.**

1. Data Import

Accurate data import was the first step in preparing the system for live usage.





Steps Taken:

1. **Imported Test Data**
    - Test **student records** were imported, including Roll Number, Name, Email, and Department association.
    - Test **department records** were imported, including Department Name and Head of Department details.

2. **Verification of Relationships**
    - Once records were imported, **lookup relationships** were validated.
    - For example, each request record was checked to ensure it linked correctly to a **Student record** and a **Department record**.
    - Data integrity was confirmed by running test reports and verifying record visibility for both students and department heads.

2. Deployment

After preparing and validating the data, the next step was to move all system configurations from **Sandbox** to **Production**.

Steps Taken:

1. **Change Sets Used**
    - An **Outbound Change Set** was created in Sandbox, which included:
        - Custom Objects (Request, Department, Student).
        - Custom Fields and Relationships.
        - Flows and Process Builders.
        - Apex Classes and Triggers.
        - Lightning Web Components (LWCs).

- The Change Set was then uploaded to Production and validated before deployment.

2. **Production Deployment**

   - Once validation was successful, the Change Set was deployed into Production.
   - Admin users verified that the deployment included all objects, logic, and UI components as expected.

3. Backup & Recovery

To safeguard against data loss or corruption, a **backup and recovery strategy** was implemented.

Steps Taken:

1. **Weekly Backups**

   - A scheduled **weekly export** of critical request data was configured.
   - Backup files were stored securely in Salesforce and also mirrored to external storage.

2. **Record Ownership Mapping**

   - After deployment, ownership of records (students, requests) was carefully mapped to the correct users and departments.
   - This ensured that students could only view **their own requests**, and department heads could only view requests **assigned to their department.**

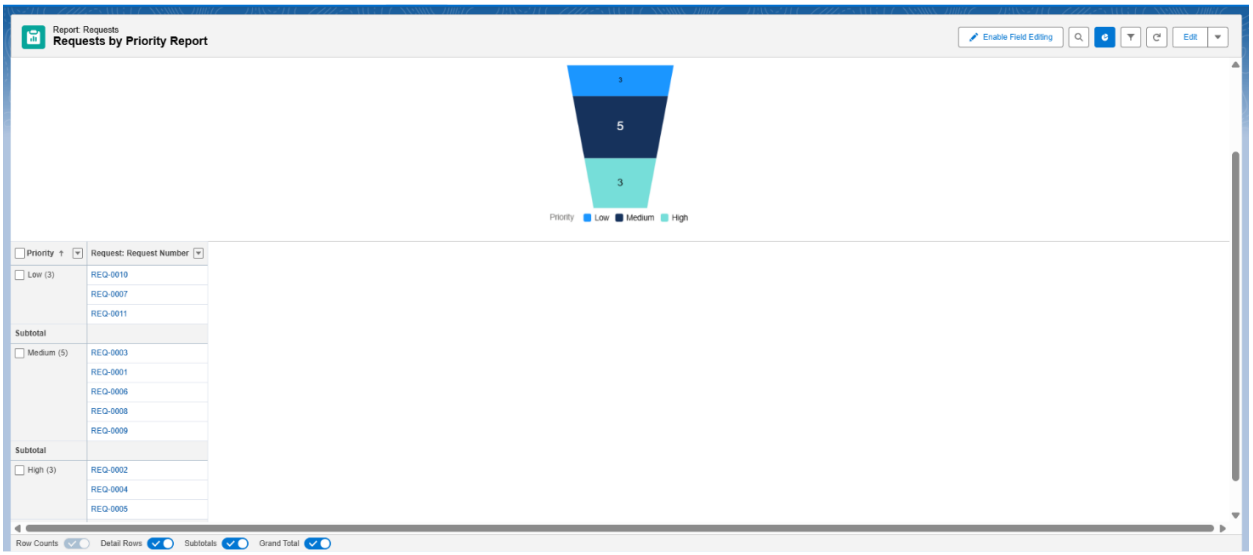
Phase 9: Reporting, Dashboards & Security Review

1. Reports

A total of **six key reports** were developed to provide comprehensive analytics

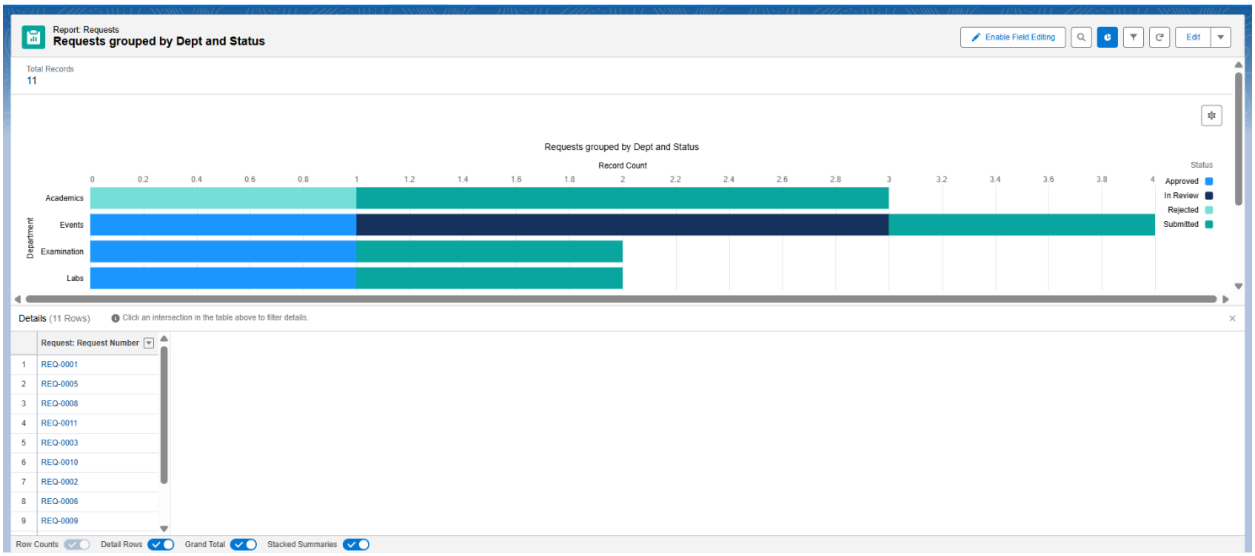
1. Requests by Priority Report

- Groups all requests based on their assigned priority (High, Medium, Low).
- Allows department heads and admins to quickly identify **critical requests** that need immediate attention.



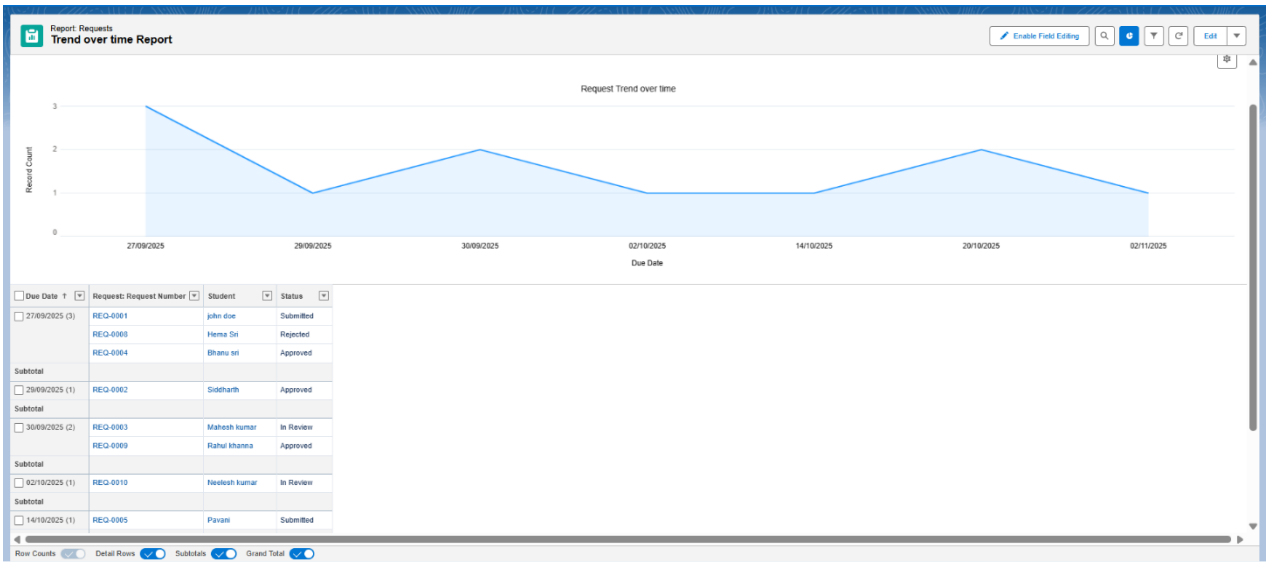
2. Requests Grouped by Department and Status

- Provides a **matrix view** showing how many requests each department has, broken down by status (Pending, Approved, Rejected, Escalated).
- Helps department heads monitor workload and track processing efficiency.
- Supports conditional formatting to highlight delayed or escalated requests.



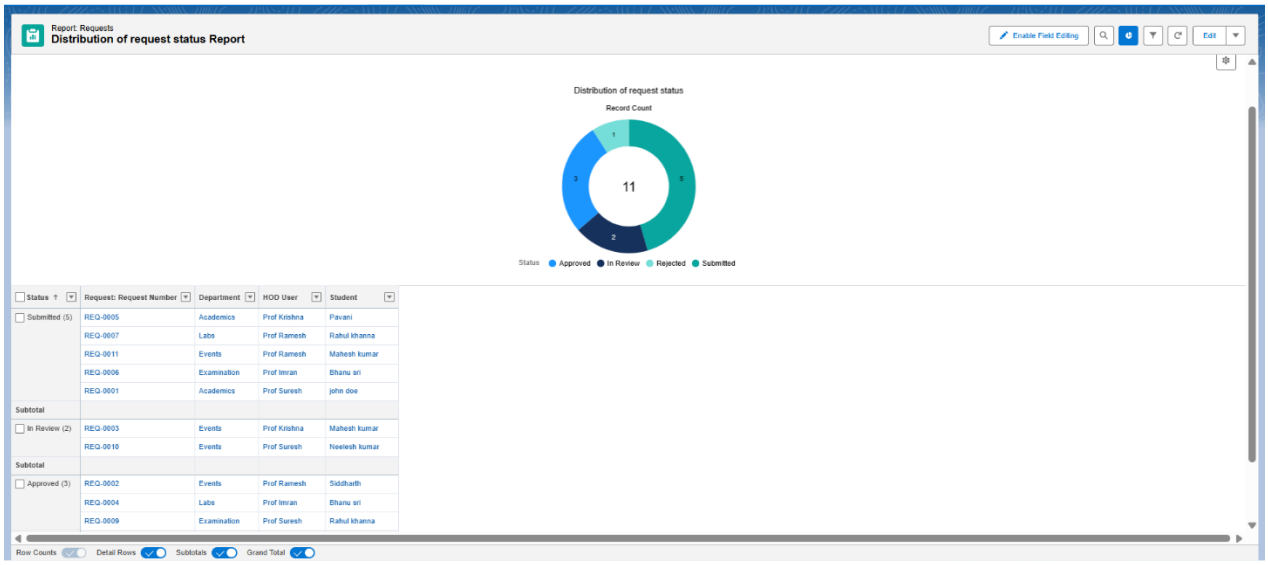
3. Trend Over Time Report

- Displays the number of requests submitted, approved, or escalated **over weekly or monthly intervals**.
- Highlights trends in request volume and approval times.
- Useful for **capacity planning** and identifying peak request periods.



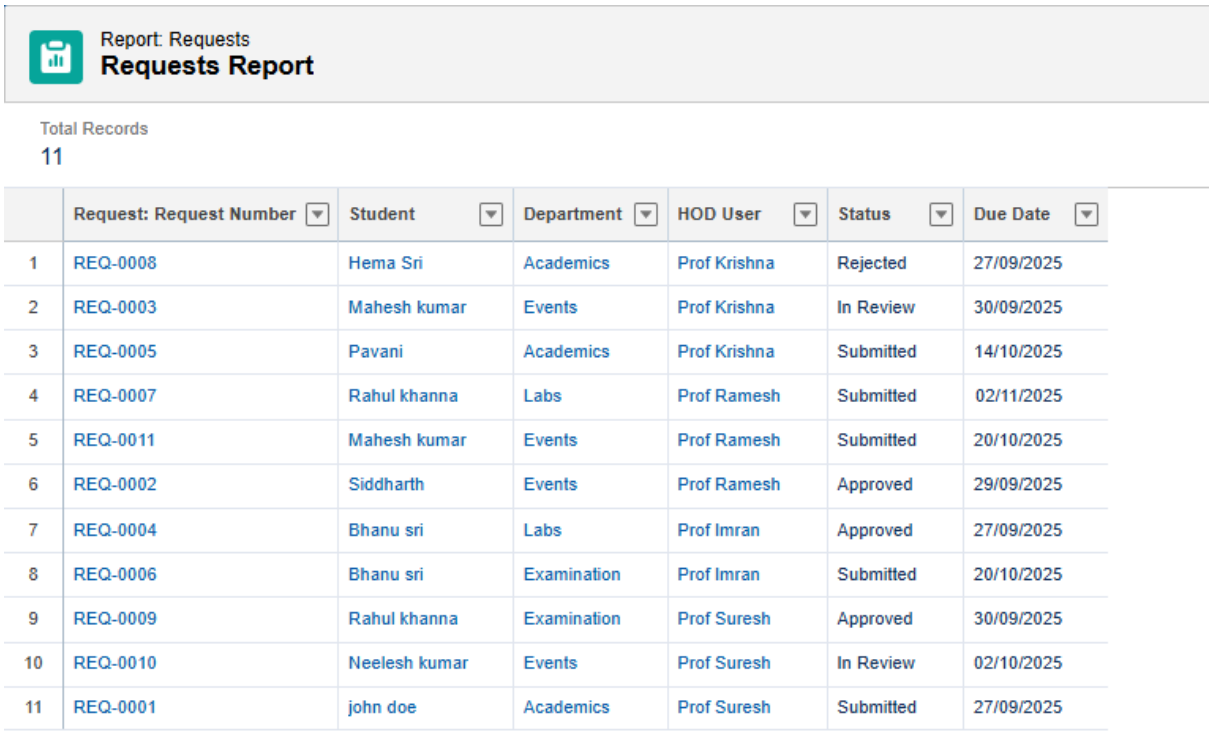
4. Distribution of Request Status Report

- Pie or donut chart showing the **percentage of requests** in each status category.
- Enables admins and department heads to quickly assess the **overall health of request processing**.



6. Requests Report Table

- Tabular report of all requests with detailed fields such as **Request Type, Status, Student, Department, Submission Date**.
- Serves as a **comprehensive log** for auditing and manual verification.
- Can be exported to CSV for offline analysis or compliance purposes.



Report: Requests
Requests Report

Total Records
11

	Request: Request Number	Student	Department	HOD User	Status	Due Date
1	REQ-0008	Hema Sri	Academics	Prof Krishna	Rejected	27/09/2025
2	REQ-0003	Mahesh kumar	Events	Prof Krishna	In Review	30/09/2025
3	REQ-0005	Pavani	Academics	Prof Krishna	Submitted	14/10/2025
4	REQ-0007	Rahul khanna	Labs	Prof Ramesh	Submitted	02/11/2025
5	REQ-0011	Mahesh kumar	Events	Prof Ramesh	Submitted	20/10/2025
6	REQ-0002	Siddharth	Events	Prof Ramesh	Approved	29/09/2025
7	REQ-0004	Bhanu sri	Labs	Prof Imran	Approved	27/09/2025
8	REQ-0006	Bhanu sri	Examination	Prof Imran	Submitted	20/10/2025
9	REQ-0009	Rahul khanna	Examination	Prof Suresh	Approved	30/09/2025
10	REQ-0010	Neelesh kumar	Events	Prof Suresh	In Review	02/10/2025
11	REQ-0001	john doe	Academics	Prof Suresh	Submitted	27/09/2025

2. Dashboards

A single **comprehensive dashboard** was created to consolidate all reports and provide an **overall view of performance**.

Dashboard Components:

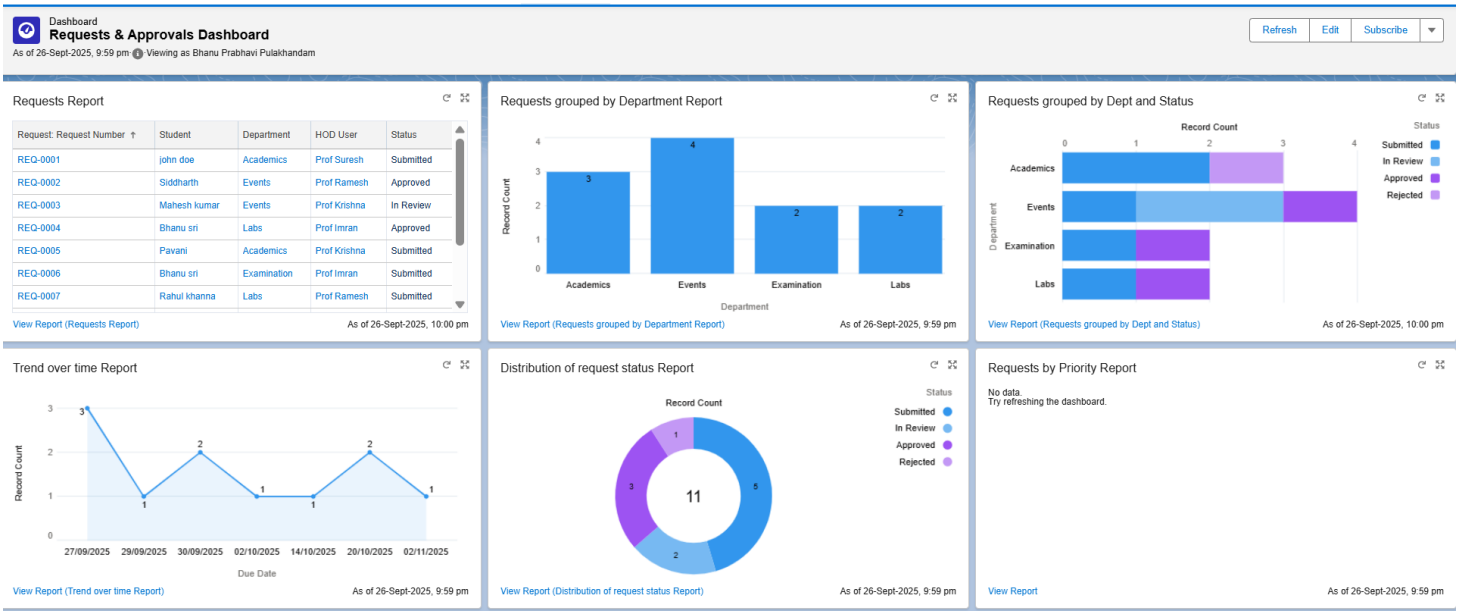
1. **Priority Overview Component**
    - Based on the *Requests by Priority Report*.
    - Displays color-coded metrics for High, Medium, and Low priority requests.
2. **Department Performance Component**
    - Based on *Requests Grouped by Department and Status Report*.
    - Pie charts highlight pending vs approved requests by department.
3. **Trend Analysis Component**

- Based on the *Trend Over Time Report*.
- Line chart showing submission trends for quick identification of peak periods.

4. **Status Distribution Component**
   - Based on the *Distribution of Request Status Report*.
   - Provides an instant snapshot of overall workflow health.

5. **Department Request Volume Table**
   - Based on *Requests Grouped by Department Report*.
   - Gives admins a clear view of workload per department.

6. **Detailed Requests Table**
   - Based on *Requests Report Table*.
   - Allows filtering and sorting for ad-hoc analysis.



3. Security Review

Ensuring proper access to reports and dashboards was crucial.

Implemented Measures:

1. **Role-Based Access**
   - Students can view only **reports of their own requests**.
   - Department Heads see **reports and dashboards for their department only**.
   - Admins have full visibility across all departments.

2. **Profile & Permission Sets**
    - Permissions configured to allow access to only relevant reports, dashboards, and underlying objects.
3. **Sharing Rules & OWDs**
    - Organization-Wide Defaults set to private for Request and Student objects.
    - Sharing rules applied to allow department heads visibility into their assigned requests.

Phase 10: Quality Assurance Testing

1. Test Cases

Comprehensive test cases were created to cover every functional aspect of the system.

Key Test Areas:

1. **Request Submission**
    - Students can submit **Leave, Event, Lab, or Project requests**.
    - Verified mandatory fields (Name, Department, Request Type, Notes).
    - Confirmed record creation triggers flows and assigns the request correctly.
2. **Auto-Assignment Flow**
    - Tested Record-Triggered Flow to automatically assign requests to the correct department.
    - Verified correct assignment for all request types.
    - Confirmed handling of bulk request submissions.
3. **Approval and Rejection**
    - Validated department head actions for approving or rejecting requests.
    - Checked status updates and email notifications to students.
4. **Email Notifications**

- Verified timely delivery of email notifications for approvals, rejections, and escalations.
- Tested different email templates for correctness.

5. **Escalation Rules**
   - Simulated requests pending >48 hours.
   - Verified that the system automatically escalated the requests and notified administrators.

2. User Acceptance Testing (UAT)

After internal QA, **students and department heads** tested the portal to validate usability and workflow efficiency.

Activities Conducted:

- Students submitted test requests and tracked status updates.
- Department heads approved, rejected, and escalated requests in real-time.
- Users tested navigation, dashboards, and report access.
- Feedback on minor UI improvements, flow adjustments, and notifications was collected.
-

3. Bug Fixes

- Fixed issues in flow auto-assignment.
- Resolved email notification delays.
- Ensured batch Apex jobs run as scheduled.

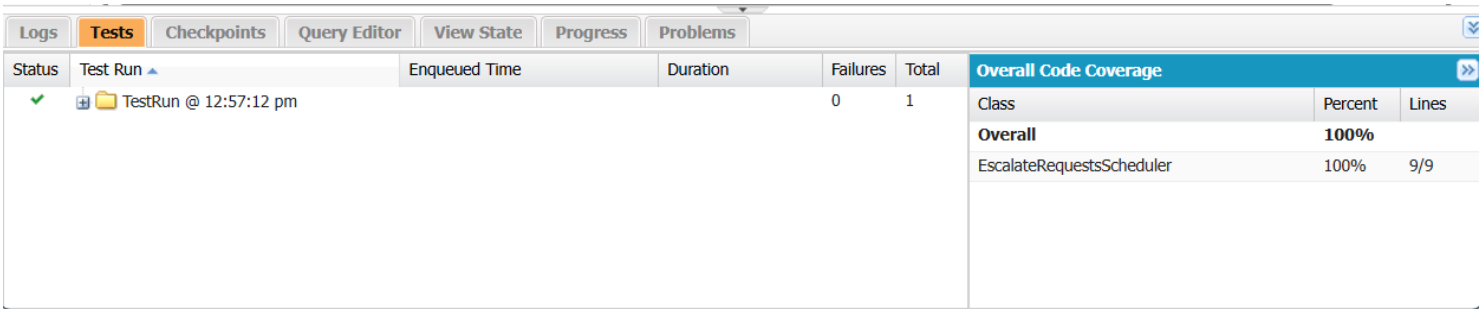
4. Final Verification

A **final verification** was conducted to ensure system readiness:

- **End-to-End Functionality**
  - Student request submission → auto-assignment → approval/rejection → email notification → escalation.
- **Reports & Dashboards**

- Verified dynamic updating of all reports and the consolidated dashboard.
- Ensured metrics reflect real-time data across departments.

- **User Roles & Security**
  - Checked role-based access for students, department heads, and admins.
  - Confirmed sensitive data is visible only to authorized users.



Executive Summary

Conclusion

The **Student Help & Permission Portal** was designed and implemented on Salesforce to streamline the request management process between students and departments. The project successfully advanced through structured phases, including **Org Setup, Data Modeling, Process Automation, Apex Programming, User Interface Development, Integration, Data Management, Reporting, and Quality Assurance Testing**. Each phase contributed to building a robust solution where students can easily submit and track requests, department heads can efficiently review and approve them, and administrators can monitor institutional performance through real-time dashboards. Automation ensures timely routing and escalation, while custom Lightning components enhance usability. With secure deployment, data integrity, and extensible architecture, the portal transforms the traditional request process into a **centralized, digital, and user-friendly experience**.

Future Scope

The portal has the potential to evolve into a **comprehensive digital campus platform** through:

- **Mobile App Integration** for anytime access.
- **AI-Powered Prioritization** and predictive dashboards.
- **Chatbot & Voice Assistant** support for FAQs and guided submissions.

- **ERP/HRMS Integration** to unify institutional systems.
- **Enhanced Security & Compliance** (MFA, GDPR/FERPA).
- **Cross-Department Expansion** into Finance, Library, and Administration.
- **Gamification Features** to boost engagement and responsiveness.

Recommendations

To ensure sustainability and continuous improvement, the following are recommended:

- **Regular Monitoring & Maintenance** with backups and audits.
- **User Training & Documentation** for students, staff, and admins.
- **Feedback Loop** to gather insights for iterative enhancement.
- **Scalability Planning** to support higher data and user volume.
- **Disaster Recovery Strategy** for business continuity.

- P. Bhanu Prabhavi